

Alfred Zimmermann, Alexander Rossmann (Hrsg.): Digital Enterprise Computing 2015,  
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2015 221

---

# Application of Process Mining for Improving Adaptivity in Case Management Systems

Eberhard Heber<sup>1</sup>, Holger Hagen<sup>2</sup> and Martin Schmollinger<sup>3</sup>

**Abstract:** The character of knowledge-intensive processes is that participants decide the next process activities on base of the present information and their expert knowledge. The decisions of these knowledge workers are in general non-deterministic. It is not possible to model these processes in advance and to automate them using a process engine of a BPM system. Hence, in this context a process instance is called a case, because there is no predefined model that could be instantiated. Domain-specific or general case management systems are used to support the knowledge workers. These systems provide all case information and enable users to define the next activities, but they have no or only limited activity recommendation capabilities. In the following paper, we present a general concept for a self-learning system based on process mining that suggests the next best activity on quantitative and qualitative data for a given case. As a proof of concept, it was applied to the area of insurance claims settlement.

**Keywords:** Adaptive Case Management, Process Mining, Business Process Management.

## 1 Introduction

In the past decade, Business Process Management (BPM) [VHW03] [We07] [DLMR13] gained more and more importance for companies due to a rising need for quickly adapting a company's processes to new business models and its requirements. The development of modeling notations like Business Process Model and Notation (BPMN) [OMG11] that can be used for business modeling as well as for execution on process engines of BPM systems [Ka95] [Ch06] [DLMR13] improves the ability of IT departments to reduce the time for automating new processes.

The implementation of processes using a BPM system is the best approach for processes for which it is possible to design a standardized model that is completely deterministic and can be reused for each process instance [SF07]. The model contains all possible activities and events and all their possible orders of execution. Process participants have to deal with the same activities in each process instance.

Experiences from BPM projects in the last years showed that this approach is not

---

<sup>1</sup> Novatec Consulting GmbH, Dieselstr. 18/1, 70771 Leinfelden-Echterdingen, eberhard.heber@novatec-gmbh.de, Germany

<sup>2</sup> Novatec Consulting GmbH, Dieselstr. 18/1, 70771 Leinfelden-Echterdingen, holger.hagen@novatec-gmbh.de, Germany

<sup>3</sup> Reutlingen University, Herman Hollerith Zentrum, Alteburgstr. 150, 72762 Reutlingen, bpm@reutlingen-university.de, Germany

applicable for all processes in all domains. Especially processes in knowledge-intense working domains, like e.g. incident management or handling of service requests, cannot be modeled and automated in this way. The reason for this is that the required activities and their sequence of execution for handling a certain process instance depends on its specific situation. Depending on this, an individual plan has to be developed involving an individual set of participants and activities. This plan is not static and can be adapted to new situations during the handling of the instance. The plan is created and adapted by a knowledge worker and after completion of the process instance stored for potential reuse on new instances [Sw10]. A knowledge worker "... is someone who knows more about his or her job than anyone else in the organization" [Dr59] and is therefore the only one who is able to develop an individual plan for solving the current process instance.

In this context the term process instance from BPM does not fit, because there is no predefined model that could be instantiated. Instead of this, the term case is preferable because of the data-centric and goal-driven nature of the work. According to [Da05] the percentage of knowledge workers in companies is between 25% and 50% of the general workforce. [Sw10] states that knowledge workers are possibly spending 95% of the workday performing knowledge work. That means, that a huge part of a company's business processes is knowledge-intense and stresses the importance of a successful case management and the need for an optimal IT support.

The BPMN standard defines ad-hoc processes that could help in such situations with the restriction that the potential number of activities in future for handling the case successfully is known at the time of modeling. Unfortunately, the implementation of ad-hoc processes in BPM systems is very seldom. Currently, this statement is also valid for CMMN [OMG14], OMG's new modeling standard for case management. This notation standardizes the graphical modeling of case management processes. Nevertheless, also this approach only helps in situations, where all possible future activities are well known. Independent of this restriction, there are currently nearly no relevant systems on the market that implement the notation. These are rather new developments which will improve in future, but do not help for the moment due to the existing restrictions concerning support for knowledge-intense processes.

Traditionally, systems that support knowledge workers got well-known as case management systems. In the last years, these systems improved their capabilities concerning the adaptivity of the case handling leading to the term *Adaptive Case Management (ACM)* [Fi15] [LM11] [Pu10]. Vendors quickly adapted this term, but each of them had a different understanding of the term adaptive in this context.

According to [LM11] a case is data-centric and the corresponding case folder includes all the documents, data, collaboration artifacts, activities, workflows, policies, rules, analytics and other information needed to handle the case. The handling itself is driven by outside events and requires incremental and progressive responses from the knowledge workers of the respective business domain with a certain goal in mind.

[KS10] demands that ACM systems provide the functionality to reuse fragments of the handling of former solutions to future cases. One proposal is to introduce templates (predefined cases containing certain case folders) that help to accelerate the start-up of a new case. After initializing a case using a template, the configuration can be adapted to the current needs of the case. After a successful handling of the case, it can be stored as a new template for future cases. Further, they want to enable the user to adapt a case at runtime to its individual needs. That means that a user is able to add, delete or change components (e.g. activities or documents) of the corresponding case folder. Furthermore, the adaptations should not be limited to be case-specific. Also case-comprehensive adaptations are possible; e.g. the adaption of global activities or documents used in all cases.

[Kr15] presented another categorization concerning the degree of adaptivity in ACM systems (see Tab. 1).

Degree of Adaptivity	Languages & Systems	Description
Adaptive	Ontologies, semantic models	On the fly adaption, self-learning, -adaption, knowledge inference, non IT-centric vocabulary
Guiding	Social BPM, Collaborative Decision Making, Integration of statistical means	Recommendation system-like, still finite set of a priori defined activities
Dynamic	BPMN, CMMN, ad-hoc tasks	Choice among predefined number of activities
Predefined	BPMN and BPM systems	Static workflows, deterministic automata, changeable through IT only

Tab. 1: Degree of Adaptivity in ACM according to [Kr15].

The execution of BPMN models on process engines has the lowest degree of adaptivity and is categorized as *Predefined*. The next degree of adaptivity, called *Dynamic*, is reached if the pre-mentioned ad-hoc processes are realized, e.g. using BPMN or CMMN on a respective BPM or ACM engine. The restriction here is that all potential activities have to be known at modeling time. One step further, more adaptivity can be received in systems that have knowledge about the behavior of other users in the same situation. These approaches are categorized under *Guiding* and are derived from web 2.0 ideas and recommendations systems of internet shops (“people that bought this product were also interested in these products”). The resulting area in BPM is called Social BPM. This approach can be very helpful for knowledge workers, because they show potential plans or activities that have been used successfully for comparable cases in history. The fourth level, called *Adaptive*, describes an approach based on semantic technologies. The idea is to model and represent the knowledge of the given domain as an ontology. Ontologies help systems to infer plans or future activities automatically applying the ontology on the case folder. Knowledge workers decide whether to use the recommendation or to prefer

an own plan or activity. Dependent on the decisions of the knowledge workers the underlying ontologies must be adapted in order to improve future recommendations. Since the ontology represents knowledge, the knowledge worker (and not the IT department) should be able to extend it in cases where unpredicted situations arise.

The last two approaches are similar in their attempt to learn from the decisions of the knowledge workers. While the first one (*Guiding*) derives the recommendation from data that was collected during the handling of former cases, the latter approach (*Adaptive*) manages an explicit model that contains information about entities and their relations in the respective domain.

In this paper, we present a concrete concept for realizing adaptivity in case management systems based on historic case handling data. The concept can be ranked in-between the adaptivity degrees (Tab. 1) *Guiding* and *Adaptive*, because it has properties of both categories. The concept can be characterized as follows:

- No explicit rule management: In general, the development of rule-based systems is very work-intensive, because you have to work out how the business is really done. Besides political issues, you are also dependent of employees explaining their daily business from their personal point of view and you have to gather this information together and to extract a set of rules. Further, a management of these rules has to be installed, i.e. a process for adding, deleting or updating rules. In our approach, we don't use explicit business rules. Instead, we use the collected data of previous handled cases that implicitly contains business rules.
- Process mining Technology: The existing domain-specific case management systems store each step of handling a case in their log files. Using process mining technology helps to discover the real process model out of the provided log files. This model is the base for the system's recommendations. Although knowledge workers should be able to decide the next steps just on base of the current case folder, the information about decisions and their consequences in similar former cases is valuable. Even if it is not possible in the respective knowledge-intense process to adopt the recommendations of the system directly, the knowledge worker should be able to revise them successfully for his current case.
- Self-learning: Each handled case leaves its footsteps in the log files of the case management systems. By continuously updating the data store for process mining with new handled cases the system learns with each of them.
- Quantitative and qualitative data: Besides data that just describes the quantitative appearance of certain sequence of activities, we also want to include qualitative data like the amount of damage and the duration of settlement in order to give users a domain-specific orientation for their decision.
- Add-on or plugin character: The concept can be realized as an add-on module to existing domain-specific case management systems. In real world scenarios, the existing software is very complex, closed and monolithic. Usually, only data

integration approaches are acceptable concerning costs. This minimalistic kind of integration is sufficient for our concept.

Although the approach is very generic, we applied it to the area of insurance claim management, where we could work on (in part anonymized) real world data as a base for our prototypic implementation. Before we go into details of the concept, we sketch the application domain and motivation.

## **2 Practical Motivation**

In the field of insurances, claim settlements are handled as cases. The handling of these cases can only be standardized in very rare situations, e.g. breakage of car glass. In the most cases, we are confronted with knowledge-intensive processes. The clerk of the insurance acts as a knowledge worker and must handle the cases in an interactive and event-driven way, because new situations may arise every time like e.g. a letter from the lawyer of a client or the unforeseen behavior of an opposing insurance. In practice, that means e.g. that the number and kind of activities necessary for settling a claim from a scratch in a car's varnish is totally different from those needed for handling a write-off car accident.

In this paper, we assume that claim settlements of the insurance is implemented by a domain-specific respectively an insurance case management system. The system stores cases and case-specific activities that can be evaluated afterwards. On top of this, we want to provide a recommendation system to support the clerks while they are handling their cases one after each other. On base of former cases, the system suggests the next best activity (NBA) for a certain case. Depending on the current case state (given by the last activity or event) the system calculates the NBA by taking into account the number of occurrences of a certain activity in former cases in the same state, the resulting amount of damage and the time needed to finally settle the claim for taking this activity. Ideally, we don't recommend only one opportunity but a list of opportunities prioritized by the just mentioned figures.

For our experiments, we use a data pool from an insurance company containing about 25.000 cases in 257.000 data sets.

## **3 A Concept for a Recommendation System based on Process Mining**

Process mining technology [VdA11] plays a key role in our concept. Process mining can produce process models using activity data captured in event logs of information systems reflecting the real process flow. [Br15] suggests four use cases for process mining in ACM systems.

First, process mining can make routine processes more effective and efficient what will also help knowledge workers in their daily work, because they also use routine processes. Second, process mining discovers hidden routine processes in the work of knowledge workers. These can be automated using process engines improving the overall performance of a knowledge worker. Third, process mining techniques can be used to recapture the behavior of knowledge workers. This transparency can be used in training reviews to help knowledge workers to improve their personal performance. Finally, Process Mining helps to bridge the gap between structured (BPM) and unstructured work (ACM) in an end-to-end process by enabling a feedback and align mechanism for the structured part and making unstructured parts more transparent. In addition to these suggestions, we use the mined graph for realizing a self-learning recommendation system.

For each case, we have a sequence of activities stored in our data pool. The data pool is an aggregation of the log files of the domain-specific case management system. Using process mining algorithms, we create a directed graph containing all sequences of all cases, where the nodes of the graph represent activities and the directed edges between two nodes implies that one activity sequentially followed the other in at least one case. Tab. 2 shows a simplified data pool of cases. One activity in the sequence of a case is represented by an upper case letter.

Case	Sequence of Activities	Amount of Damage	Duration
1	ABCDE	\$1.000,00	5 days
2	ACBDE	\$500,00	5 days
3	ABCDDE	\$1.500,00	10 days

Tab. 1: Simplified sample cases, activities and qualitative figures from the data pool.

The resulting graph is the base for calculating the list of potential NBAs described earlier. If we want to know the set of activities users had chosen in former cases, we look up the node in the graph that represents the latest activity of our current case (current node). All nodes that are reachable by an adjacent edge represent the set of possible NBAs. Fig. 1 shows the mined graph for the data pool of Tab. 2. Let us assume that the last activity of a current case was C, then the graph in Fig. 1 tells us that the set of possible NBAs contains the activities B and D.

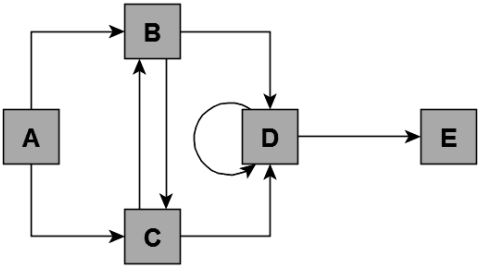


Fig. 1: Mined graph for the cases in Tab. 2 without quantitative or qualitative figures.

During the creation of the graph, we calculate quantitative and qualitative figures that can be used for prioritizing the list. A first obvious figure is quantitative, namely the number of subsequent occurrences of two activities in the activity sequences of all cases. The more often a certain edge in the graph was taken in the past the more probable it is that this could also be a good suggestion for the current case.

Since this is not always true, we add qualitative factors to improve the recommendation. For each case, we have the amount of damage (real costs after closing the case) and the time needed to settle the claim stored in our data pool. During the creation of the graph, we calculate for each edge of the graph the average costs produced and time needed for all cases that passed the edge. Of course a clerk wants to manage the claim settlement as fast and as cheap as possible, hence, these two figures help to orientate. In Fig. 2, we sketch the resulting graph for the cases in Tab. 2 with quantitative and qualitative figures.

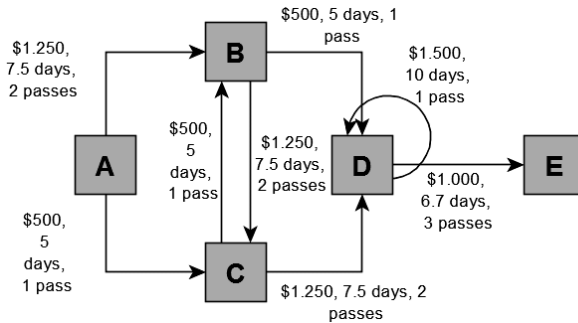


Fig. 2: The mined sample graph for the cases in Tab. 2 with quantitative and qualitative figures.

Without explicit explanation, we used the basic ideas of the well-known  $\alpha$ -algorithm [VWM04] for mining the graph in Fig. 1 from the data pool sketched in Tab. 2. This is a discrete process mining algorithm that includes every activity and sequence as it is represented in the data pool. Applied to real-world data this may lead to two main problems well-known in the area of process mining – noise and complexity.

The term noise is an analogy to problems in telecommunication, where sound waves are distorted and can hardly be interpreted and understood. In our case it stands for inconsistent or incorrect data sets. E.g. missing start, intermediate or end activities or wrong assigned activities in the sequences of the cases. The term complexity addresses the problem that real-world data is very fine-grained. Using process mining algorithms similar to the  $\alpha$ -algorithm can lead to high-resolution graphs that contain every activity and edge although it is statistically irrelevant. This makes it difficult to handle the graph, which means it is more difficult and costly to visualize and to analyze it. The view to the essential is barred.

To avoid these problems, we used the fuzzy mining algorithm [GW07]. This algorithm uses significance and correlation metrics that can be configured in a way that a graph in

the desired granularity is mined. The idea of the algorithm can be compared to the zoom level in online maps. The lower the zoom level the less information is contained in the resulting map. The fuzzy algorithm provides the quantitative figures as described above out of the box. In order to include the qualitative figures, we equipped the algorithm with additional information about minimal, maximal and average amount of damage, average duration, as well as the number of passes. All these information is stored at the edges during the mining of the graph. By that it is possible to evaluate each adjacent edge from the point of view of a certain node and to generate the desired prioritized list of activities.

The prioritization function maps the input values to a scale from 0 to 10, where 10 is the highest priority. As already explained, the function calculates a value for each edge using the stored information at the edge, namely costs, duration and number of passes. Since in first order, insurances want to work cost efficient the costs incur with a double weight in the calculation.

Applying this function to each edge leads to a sorting criteria for the list of possible next activities; the activities are sorted in descending order with respect to calculated priority. The sorted list containing all possible next activities and their quantitative and qualitative figures can be presented for inspection to the clerk within the application. The NBA is the activity in the list that has the highest priority (of course, sometimes there may be more than one activity with the highest priority). The clerk in the role of a knowledge worker is free to choose one alternative from the list or to decide to do something totally different. Independent of this decision, the clerk executes the next activity using his casual insurance case management system. This decision is incorporated in future recommendations by updating the data pool of our NBA system.

## 4 Prototypic Implementation and Performance Issues

We realized a web application prototype using the Java EE platform [DS13]. The process mining part was realized using the Java-based generic open source framework ProM published under the GPL license (v2). ProM is a platform for process mining, analysis and conversion that can be extended by plugins. Particularly, it provides several plugins for process mining algorithms. We integrated the fuzzy mining algorithm in our prototype by using and adapting the source code of the respective ProM plugin<sup>4</sup>.

Further, the prototype realizes the access to our real-world data pool, a user interface that allows selecting subsets of the cases for claim classes (e.g. comprehensive cover or third party liability) stored in the data pool and the mining of the decision graph. For simulation reasons, the creation of new cases, the calculation of the prioritized NBA list as describe in the last section, the creation of new activities instead of just selecting them

---

<sup>4</sup> Both documentation and software (including the source code) can be downloaded from [www.processmining.org](http://www.processmining.org)



from the suggested list and the inclusion of new handled, finalized cases in the decision graph is realized, too. The latter data is just stored in main memory and hence volatile. With these features it is possible to simulate the handling of cases by using the recommendation system.

During our simulations, we recognized serious performance problems. Experiments on a casual developer computer showed that the prototype is only able to extract and analyze about 50 cases out of 25.000 in an acceptable time. Although in a productive environment the processing would be done on a more performant server, this seems to be a bottle neck for the whole approach. Particularly, if the software is used in a multi-user environment, we would have to generate multiple graphs at a time on the server and all of them have to be held in memory for a certain time (scope). One technical approach is to replace the persistence layer of the application. The prototype uses Java's standard O/R mapper JPA. Though the usage of standard technologies has often big advantages, in this case it was not an adequate choice and provides room for improvement. The performance issue needs further investigations with respect to technology, algorithms and data structures.

## 5 Conclusions

In the last sections, we explained the characteristics of knowledge-intense processes and the need for IT systems that support knowledge workers without limiting their freedom of decision. In this context, we presented an approach to increase the degree of adaptivity in domain-specific case management systems. The core of the presented concept is the usage of process mining algorithms for a self-learning recommendation system that can be realized as an add-on to the given case management system. The recommendation system suggests the next best activity on quantitative and qualitative data for a given case. As a proof of concept, it was applied to the area of insurance claims settlement. The resulting prototype was used for simulations on real-world data from an insurance. During the simulations, serious performance problems were recognized that need further investigation. Beside of this issue, the simulations were promising, but have not yet been verified by a professional insurance clerk. This is the second open question. Are the recommendations generated by the system really helpful? In order to answer this question, the prototype has to be improved in functionality and performance, the data pool has to be updated. Afterwards, a promising idea is that test clerks from an insurance use the prototype in parallel to their normal software environment in daily business. Ideally, for each case they decide on base of their normal environment what to do next. Before they commit their decision in the insurance's system, they also ask our prototype what to do next and create a protocol about the degree of conformity of both decisions. Afterwards the protocols and interviews with the test clerks will hopefully give new insights for improving the quality of this approach.

## References

- [Br15] Brantley, W. A.: Ontology-based ACM – The Need for Truly Adaptive Systems. In: Fischer, Layna et al.: *Thriving on Adaptability - Best Practices for Knowledge Workers*. Future Strategies, 2015.
- [Ch06] Chang, J. F.: *Business Process Management Systems*. Auerbach Publications, 2006
- [Da05] Davenport, T. H.: *Thinking for a Living – How to Get Better Performances and Results from Knowledge Workers*. Boston: Harvard Business School Publishing, 2005.
- [DLMR13] Dumas, M.; La Rosa, M.; Mendling, J.; Reijers, H.: *Fundamentals of Business Process Management*. Springer, 2013.
- [DS13] DeMichiel, L.; Shannon, B.: Java Community Process: JSR 342: Java™ Platform, Enterprise Edition 7 (Java EE 7) Specification, URL <https://jcp.org/en/jsr/detail?id=342>, 2013.
- [Dr59] Drucker, P. F.: *Landmarks of Tomorrow*, 1959.
- [GW07] Günther, C. W.; van der Aalst, W.: Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metric. In: *Proceedings of the 5th International Conference Business Process Management (BPM)*, LNCS Volume 4714, pp 328-34, Springer, 2007.
- [Ka95] Karagiannis, D.: BPMS: Business Process Management Systems. In: *ACM SIGOIS Bulletin* 16, pp 10-13, 1995.
- [Kr15] Kress, J.; Utschig-Utschig, C.; Normann, H.; Winterberg, T.: Ontology-based ACM – The Need for Truly Adaptive Systems. In: Fischer, Layna et al.: *Thriving on Adaptability - Best Practices for Knowledge Workers*. Future Strategies, 2015.
- [KS10] Khoyi, D.; Swenson, K. D.: *Templates, Not Programs*. In: *Mastering the Unpredictable – How Adaptive Case Management will Revolutionize the Way that Knowledge Workers get Things done*. Meghan-Kiffer Press, 2010.
- [LM11] Le Clair; Miers: *The Forrester Wave: Dynamic Case Management Q1 2011* / Forrester Research. 2011.
- [OMG11] Object Management Group: *Business Process Model And Notation (BPMN) Version 2.0* / Object Management Group. URL <http://www.omg.org/spec/BPMN/2.0/PDF>, 2011.
- [OMG14] Object Management Group: *Case Management Model And Notation (CMMN) Version 1.0* / Object Management Group. URL <http://www.omg.org/spec/CMMN/1.0/>, 2014.
- [Pu10] Pucher, M. J.: *The Elements of Adaptive Case Management*. In: *Mastering the Unpredictable – How Adaptive Case Management will Revolutionize the Way that Knowledge Workers get Things done*. Meghan-Kiffer Press, 2010.
- [SF07] Smith, H. ; Fingar, Peter: *Business Process Management - The Third Wave*. Meghan-Kiffer Press, 2007.

- [Sw10] Swenson, K. D.: The Nature of Knowledge Work. In: Mastering the Unpredictable – How Adaptive Case Management will Revolutionize the Way that Knowledge Workers get Things done. Meghan-Kiffer Press, 2010.
- [VdA11] Van der Aalst, W.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, 2011.
- [VHW03] Van der Aalst, W.; ter Hofstede, A.; Weske, M.: Business Process Management: A Survey. In: Lecture notes in computer science Bd. 2678/2003. Springer Verlag, p 1019, 2003.
- [VWM04] Van der Aalst, W.; Weijters, T; Maruster, L: Workflow Mining: Discovering process models from event logs. In: IEEE Transactions on Knowledge and Data Engineering Vol. 16 (9), pp 1128-42, 2004.
- [We07] Weske, M.: Business Process Management. Springer Verlag, p 7, 2007.